IEA/AIE 2026

From Black-Box to Glass-Box: Explainable AI for Applied Intelligent Software Systems Across the Software Development Life Cycle

Thi-Hong-Cuc Le[1,2] and Xuan-Bach Le[1,2]

[1]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT)

[2]Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

Presenter: Thi-Hong-Cuc Le

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

I-SOMET

UTM UNIVERSITI TEKNOLOGI MALAYSIA

Springer

ISAI

acm SIGAI

Authors at a Glance

Xuan-Bach Le

lexuanbach.github.io

Lecturer / Co-author

RAISE Lab · Lab Head

Advisor profile

Lecturer at HCMUT working across program verification, formal logic, RAG/LLMs, program synthesis, and reliable software engineering.

Specialty: Safe LLM Agents

Thi-Hong-Cuc Le

nausmind.com

First author / Presenter

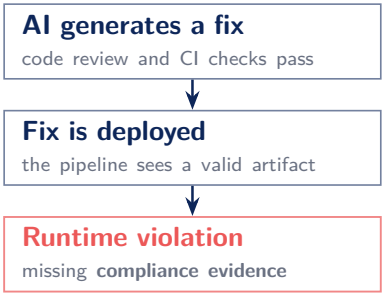
RAISE Lab · Core Member

Research focus

Presenter focusing on explainable AI, software engineering evidence, and trustworthy AI-assisted SDLC.

Agenda

Opening Scenario: A Passing Fix Can Still Fail



Core question

Can engineers justify, audit, and certify AI-assisted decisions?

Evidence gap

"Passing tests ≠ traceable engineering evidence."

Thesis: **XAI for software engineering is a lifecycle, persona, and certification problem.** (Ibrahim & Lim; Zhao et al.)

SOURCES Ibrahim & Lim, Energies 2025 ■ Zhao et al., IEEE TSE 2025

The Black-Box Problem in the SDLC

Trust

Black box

Why did AI suggest this refactor?

Glass box

Trace from requirement to code change.

Safety

Black box

LLM code passes lint but exposes a CWE risk.

Glass box

XAI flags the risk before merge.

Assurance

Black box

A plausible rationale cannot be verified.

Glass box

Provenance log becomes an auditable artifact.

The gap is not one tool. It is the lifecycle. (Huang et al.)

SOURCES Huang et al., ESWA 2024

Three Research Questions

RQ1 – Coverage

Which SDLC phases and stakeholder roles are covered in AI-assisted SE?

Motivates: phase/persona coding

RQ2 – Faithfulness

How rigorously are XAI explanations validated for causal faithfulness?

Motivates: Q2 criterion, LLM gap

RQ3 – Personas

To what extent are explanations grounded in concrete stakeholder roles?

Motivates: persona mapping

These RQs map directly to the coding scheme and quality criteria

SMS / mapping protocol: Kitchenham & Charters 2007; Petersen et al. 2015.

SOURCES Kitchenham & Charters, EBSE 2007 ■ Petersen et al., IST 2015

Why Prior Surveys Miss the Core Gap

Related work is strong *within* each lens — but no one lens links explanations to the SDLC, personas, *and* faithfulness.

General XAI (Awal & Roy)

Strong taxonomies and evaluation metrics.

× Weak linkage to SE artifacts and lifecycle decisions.

AI in SE (Ahmed et al.)

Broad coverage of code, defects, and testing.

× Explainability treated as a secondary concern.

LLM task studies (Umama et al.)

Detailed, tool-level evidence per task.

× No cross-phase or cross-persona synthesis.

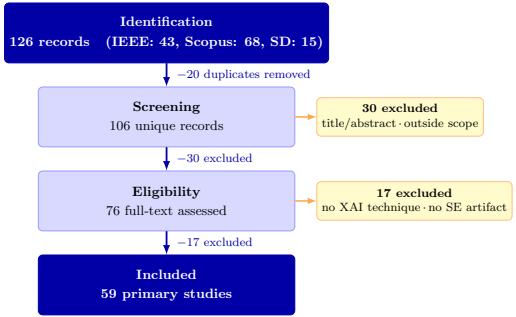
Our survey closes the gap

- ✓ **SDLC-wide** across ML, DL, and LLM tools
- ✓ **Faithfulness-coded** (Q2) — verified, not assumed
- ✓ **Persona-grounded** (Q3) SDLC evidence

Closest prior survey (Arora et al.): phase-oriented only — we add cross-generation synthesis, faithfulness coding, and persona grounding.

sources Awal & Roy, JSS 2024 ■ Ahmed et al., ACM TOSEM 2025 ■ Umama et al., IEEE Access 2025 ■ Arora et al., COMPSAC 2025

Method Rigor: A Traceable SMS



Search scope

- IEEE Xplore, Scopus, ScienceDirect
- search executed: **15 January 2026**
- **2019–2025**, English, peer reviewed

Quality assurance

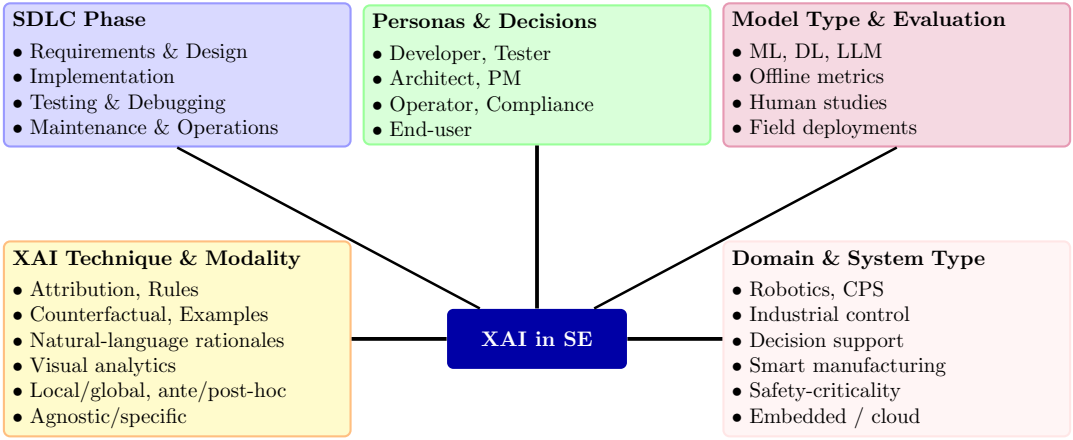
- 3 criteria: evaluation, faithfulness, persona
- inter-rater reliability: $\kappa = 0.81$ (Landis & Koch 1977)

59 studies · 4 SDLC phase groups · 3 model families

SMS protocol: Kitchenham & Charters 2007; Petersen et al. 2015; Page et al. 2021.

SOURCES Kitchenham & Charters, EBSE 2007 ■ Petersen et al., IST 2015 ■ Page et al., BMJ 2021 ■ Landis & Koch, Biometrics 1977

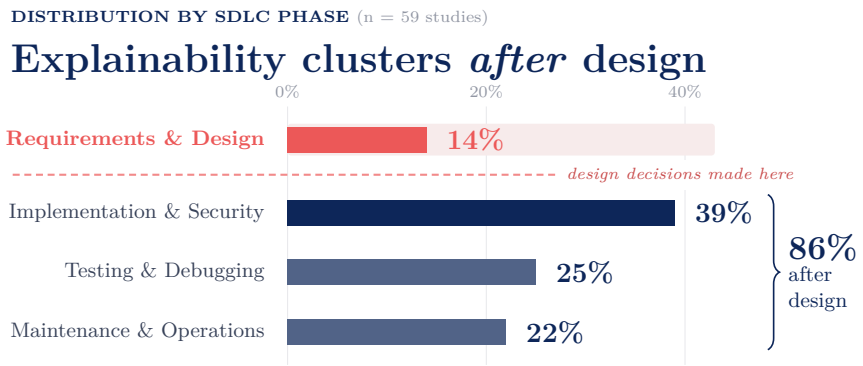
Contribution 1: Six-Dimensional Coding Framework



RQ1 → Phase & Persona **RQ2 → Faithfulness** **RQ3 → Persona & Domain**

Finding 1: Explainability Arrives Too Late

RQ1



Interpretation: requirements & design remain the least-supported SDLC phases.

Finding 2: The Faithfulness Crisis

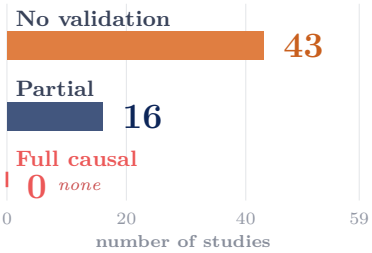
RQ2

FAITHFULNESS EVIDENCE ACROSS 59 STUDIES

Plausible explanations, unproven causally

0/59

studies with full causal validation



Partial validation

LLM 4/37

11%

Non-LLM 12/22

55%

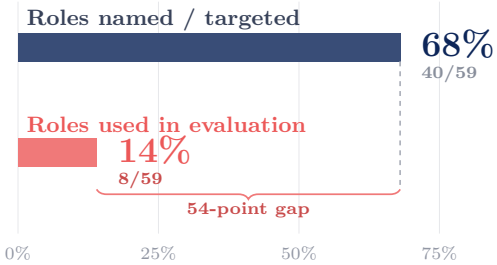
Implication: faithfulness is the weakest evidence layer for audit-ready XAI.

Finding 3: Persona Alignment Is Claimed, Not Shown

RQ3

PERSONA TARGETING VS. PERSONA-GROUNDED EVALUATION

Roles are named, not evaluated



Role-specific explanation needs

Developers — code context

Testers — traceable failure logic

Operators — drift / root-cause

Compliance — provenance logs

Gap: roles are named as motivation, but rarely evaluated as design targets.

Contribution 2: Reproducible Evidence Base (Q1–Q3 Coding)

How Findings 1–3 become reproducible evidence

Criterion	What we coded (59 studies)
Q1 Evaluation rigor	All studies report an explicit evaluation type (metrics, user study, or deployment).
Q2 Faithfulness	Causal vs. proxy validation coded as Yes / Partial / No — none reached full causal.
Q3 Persona grounding	Whether a concrete stakeholder role is operationalised in evaluation.

Findings 1–3 are measured outcomes of this coding base.

Contribution 3: Persona-to-Architecture Mapping

RQ3

Four parallel architecture patterns

Box 1 **IDE-Integrated**
Code → developer UI

Box 2 **CI/CD Services**
Tests → decision gate

Box 3 **Domain MLOps**
Req./logs → dashboards

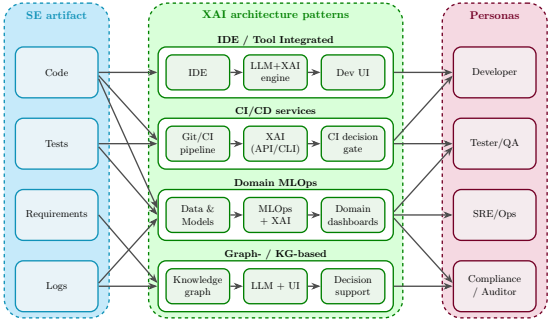
Box 4 **Graph / KG**
Logs → audit trail

Persona	Artifact	Box	Explanation need
Developer	Code	1	Inline, actionable code context
Tester / QA	Tests	1+2	Traceable test logic and CI gate evidence
SRE / Ops	Logs	3	Drift-aware root-cause dashboards
Compliance	Req.	4	Graph-based audit trails and evidence bundles

Many-to-many mapping, e.g. Tester spans Box 1 and Box 2.

Tailor explanations to who needs them, in the tool they already use.

Blueprint for Certification-Ready XAI



Embed XAI where SE decisions already happen.

Study limitations: Three databases; 59 of 126 after PRISMA; corpus skews to 2024–2025 LLM tools.

Boundary: Conceptual blueprint — industrial validation still required for certification.

Research Agenda: Three Priorities

1. Faithfulness-by-design

Embed perturbation analysis, causal probing, and static-dynamic checks into XAI workflows.

→ Governance: explanations become verifiable decision logs.

2. Persona-aware evaluation

Link explanations to stakeholder decisions, not only model metrics or subjective plausibility.

→ Quality: faithfulness moves into acceptance criteria.

3. Certification-ready pipelines

Standardise logs, provenance, uncertainty, and traceability as first-class SE artifacts.

→ Certification: aligned with ISO/IEC 42001 & 25059 (ISO/IEC 42001; ISO/IEC 25059).

From ad hoc explanations to auditable glass-box SE. (Awal & Roy)

SOURCES ISO/IEC 42001:2023 ■ ISO/IEC 25059:2023 ■ Awal & Roy, JSS 2024

Conclusion: Did We Answer Our RQs?

	Question	Answer from the mapping study	Verdict
RQ1	Where is XAI coverage?	Impl./Security **39%** vs Req./Design **14%**; dev/tester focus **68%**.	Late-skewed
RQ2	How faithful are explanations?	**0/59** full causal; LLM **4/37** vs non-LLM **12/22** partial.	Unproven
RQ3	Are personas operationalised?	Only **8/59** studies operationalise a concrete role.	Rare

Open problems

1. Faithfulness-by-design
verify explanations, do not assume them

2. Persona benchmarks
evaluate explanations by stakeholder role

3. Certification-ready
treat explanations as SE artifacts

From black-box AI assistance to glass-box software engineering (Ibrahim & Lim; Velasco et al.).

sources Ibrahim & Lim, Energies 2025 ▪ Velasco et al., ICPC 2025

TAKEAWAY

Evidence first. Verify explanations. Log for audit.

XAI in SE
Late-skewed and role-imbalanced

Faithfulness
Unproven until causally tested

Engineering practice
Treat explanations as SE artifacts

Thank You

Questions & Discussion

CONTACT

Thi-Hong-Cuc Le
lthcuc.sdh242@hcmut.edu.vn

Xuan-Bach Le
lexuanbach@hcmut.edu.vn

IEA/AIE 2026, YR-AIS Track

References

[1] Abboush, M., Ghannoum, E., Rausch, A.: An explainable hybrid deep learning-enabled intelligent fault detection and diagnosis approach for automotive software systems validation. Knowl.-Based Syst. **334** (2026)

[2] Aditama, P., Eita, M., Koziol, T., Zhukova, K., Dillen, M., Sanchez, F., Schemmer, M., Nitsche, M., Tomina, M., Kimari, T.: Deployment of an ai-based well integrity monitoring solution. In: SPE Conf. (2024)

[3] Ahi, K., Valizadeh, S.: Large language models (LLMs) and generative AI in cybersecurity and privacy: A survey of dual-use risks, AI-generated malware, explainability, and defensive strategies. In: SVCC. pp. 1–8 (Jun 2025)

[4] Ahmad, N., Zhang, C.: Interpretable vulnerability detection in LLMs: A BERT-based approach with SHAP explanations. Comput. Mater. Contin. **85**(2), 3321–3334 (2025)

[5] Ahmed, I., Aleti, A., Cai, H., Chatzigeorgiou, A., He, P., Hu, X., Pezzè, M., Poshyvanyk, D., Xia, X.: Artificial intelligence for software engineering: The journey so far and the road ahead. ACM Trans. Softw. Eng. Methodol. **34**(5) (2025)

[6] Albaroudi, E., Mansouri, T., Hatamleh, M., Alameer, A.: Saudi arabia's vision 2030: Leveraging generative artificial intelligence to enhance software engineering. In: WiDS PSU. pp. 1–6 (Apr 2025)

References

[7] Arora, L., Girija, S.S., Kapoor, S., Raj, A., Pradhan, D., Shetgaonkar, A.: Explainable artificial intelligence techniques for software development lifecycle: A phase-specific survey. In: COMPSAC. pp. 2281–2288 (Jul 2025)

[8] Awal, M.A., Roy, C.K.: EvaluateXAI: A framework to evaluate the reliability and consistency of rule-based XAI techniques for software analytics tasks. J. Syst. Softw. **217**, 112159 (Nov 2024)

[9] Baltaji, R., Thakkar, P.: Probing numeracy and logic of language models of code. In: InteNSE. pp. 8–13 (May 2023)

[10] Bello, M., Bello, R., García, M.M., Nowé, A., Sevillano-Garcia, I., Herrera, F.: A three-level framework for LLM-enhanced explainable AI: From technical explanations to natural language. Inf. Syst. Front. (2025)

[11] Berrouyne, O., El Bajta, M., Benelallam, I.: Systematic mapping study on AI integration in CI/CD pipelines. In: ICOA. pp. 1–6 (Oct 2025)

[12] Chang, X., Li, D., Wong, W.E.: Large language models for software fault localization: A survey. In: DSA. pp. 111–120 (Nov 2025)

References

[13] Esposito, M., Palagiano, F., Lenarduzzi, V., Taibi, D.: On large language models in mission-critical IT governance: Are we ready yet? In: ICSE-SEIP. pp. 504–515 (Apr 2025)

[14] Firizqi, J.D., Indrajit, R.E.: Unified DevSecOps toolchain for secure AI-enabled maritime applications: Theory, concepts, and implementation. In: ICIC. pp. 1–6 (Oct 2025)

[15] Franzosi, D.B., Alégroth, E., Isaac, M.: LLM-based labelling of recorded automated GUI-based test cases. In: ICST. pp. 453–463 (Mar 2025)

[16] Gajjar, J., Subramaniakuppusamy, K., El Kachach, N.: Malcodeai: Autonomous vulnerability detection and remediation. In: IRI. pp. 31–36 (2025)

[17] Gatchalee, P.: A model-driven and explainable framework for LLM-powered text classification in web intelligence and marketing applications. Adv. Artif. Intell. Mach. Learn. **5**(4), 4532–4552 (2025)

[18] Gutic, B., Papić, T., Dakić, P.: Software quality and compliance in intelligent health monitoring systems: A case study of Baby FM. In: CEUR-WS Proc. vol. 4077 (2025)

[19] Hegedűs, C., Varga, P.: Tailoring MLOps techniques for industry 5.0 needs. In: CNSM 2023 (2023)

References

[20] Hu, B., Tunison, P., RichardWebster, B., Hoogs, A.: Xaitk-saliency: An open source explainable ai toolkit for saliency. In: AAAI. pp. 15760–15766 (2023)

[21] Hu, B., Tunison, P., Vasu, B., Menon, N., Collins, R., Hoogs, A.: XAITK: The explainable AI toolkit. Appl. AI Lett. **2**(4) (2021)

[22] Huang, J., Liu, Y.: MLXOps4Medic: A service framework for machine learning and explainability operations in medical imaging AI development. IEEE Access **13**, 158149–158169 (2025)

[23] Huang, X., Li, R., Yang, J., Xiao, Q.: Positive-negative chain-of-thought prompting for table and text financial question answering. In: CBASE. pp. 196–199 (Oct 2025)

[24] Huang, Z., Yu, H., Fan, G., Shao, Z., Li, M., Liang, Y.: Aligning XAI explanations with software developers' expectations: A case study with code smell prioritization. Expert Syst. Appl. **238**, 121640 (Mar 2024)

[25] Ibrahim, A.A., Lim, H.K.: A deterministic assurance framework for licensable explainable AI grid-interactive nuclear control. Energies **18**(23) (2025)

[26] Irvine, P., Da Costa, A.A.B., Zhang, X., Khastgir, S., Jennings, P.: Structured natural language for expressing rules of the road for automated driving systems. In: IV. pp. 1–8 (Jun 2023)

References

[27] ISO/IEC 25059:2023: Quality model for AI-based systems (SQuaRE) (2023), standard. Geneva: ISO/IEC

[28] ISO/IEC 42001:2023: AI management system (2023), standard. Geneva: ISO/IEC

[29] Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in Software Engineering. Technical Report EBSE-2007-01, Keele University and Durham University (2007)

[30] Kozub, V., Druzhynin, V., Trufanova, D., Ihnatenko, P., Kolos, K.: Using artificial intelligence in software development processes: Achievements and challenges. Sustain. Eng. Innov. **7**(2), 463–476 (2025)

[31] Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)

[32] Madeyski, L., Stradowski, S.: Predicting test failures induced by software defects: A lightweight alternative to software defect prediction and its industrial application. J. Syst. Softw. **223**, 112360 (May 2025)

[33] Min, Z., Budnik, C.J.: Verification and validation of LLM-RAG for industrial automation. In: AITest. pp. 50–53 (Jul 2025)

References

[34] Mohammadkhani, A.H., Tantithamthavorn, C., Hemmatif, H.: Explaining Transformer-based code models: What do they learn? when do they not work? In: SCAM. pp. 96–106 (Oct 2023)

[35] Onan, A., Nasution, A.H., Celikten, T.: Toward reliable annotation in low-resource NLP: A mixture-of-agents framework and multi-LLM benchmarking. IEEE Access **13**, 211620–211644 (2025)

[36] Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T., Mulrow, C., et al.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ **372** (2021)

[37] Panagoulias, D.P., Virvou, M., Tsihrintzis, G.A.: Truth in trees: A multilayered linguistic framework for automated fact-checking with AI-driven reasoning. Procedia Comput. Sci. **270**, 1032–1041 (Jan 2025)

[38] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Guidelines for systematic mapping studies in SE. Inf. Softw. Technol. **52**(3), 249–265 (2010)

[39] Radnejad, M.: Explainable AI for identifying and managing test debt in automated testing. In: ESEM. pp. 528–531 (Oct 2025)

[40] Rejithkumar, G., Anish, P.R.: NICE: Non-functional requirements identification, classification, and explanation using small language models. In: ICSE-SEIP. pp. 284–295 (Apr 2025)

References

[41] Rodríguez-Cárdenas, D.: Towards more interpretable large language models for code. In: ACM SIGSOFT FSE. pp. 1270–1272 (2025)

[42] Salem, D.O., Alahmed, Y., Fnich, R., Mazroub, M., Fnich, M.: AI-driven continuous integration: Automating code review and deployment with LLMs. In: FMEC. pp. 268–274 (May 2025)

[43] Schiese, D., Perevalov, A., Both, A.: Towards llm-generated explanations for component-based knowledge graph question answering systems (2025), https://arxiv.org/abs/2508.14553

[44] Shah, M.B., Rahman, M.M., Khomh, F.: Towards understanding the impact of data bugs on deep learning models in software engineering. Empir. Softw. Eng. **30**(6) (2025)

[45] Sharanarthi, T.: Adaptive multi-agent AI framework for real-time energy optimization and context-aware code review in software development. In: ISCTIS. pp. 353–358 (May 2025)

[46] Sharanarthi, T., Polineni, S.: Real-time adaptive code analysis with a self-learning multi-agent framework: A retrieval-augmented reinforcement learning approach. In: ICAIDE. pp. 534–540 (May 2025)

[47] Sijwali, S.S., Colom, A.M., Guo, A., Saha, S.: Fixing performance bugs through LLM explanations. In: AITest. pp. 102–109 (Jul 2025)

References

[48] Sovrano, F., Vitali, F.: An objective metric for explainable AI: How and why to estimate the degree of explainability. Knowl.-Based Syst. **278**, 110866 (Oct 2023)

[49] Trigg, L., Morgan, B., Stringer, A., Schley, L., Hougen, D.F.: Natural language explanation for autonomous navigation. In: DASC. pp. 1–9 (Sep 2024)

[50] Umama, Danyaro, K.U., Nasser, M., Zakari, A., Abdullahi, S., Khanzada, A., Yakubu, M.M., Shoaib, S.: LLM-based code generation: A systematic literature review with technical and demographic insights. IEEE Access **13**, 194915–194939 (2025)

[51] Velasco, A., Garryyeva, A., Palacio, D.N., Mastropaolo, A., Poshyvanyk, D.: Toward neurosymbolic program comprehension. In: ICPC. pp. 377–381 (Apr 2025)

[52] Vilone, G., Sovrano, F., Lognoul, M.: On the explainability of financial robo-advice systems. In: Commun. Comput. Inf. Sci. vol. 2156, pp. 219–242 (2024)

[53] Walker, M., Forêt, J.M.: Unleashing the cognitive digital twin via semantic orchestration. In: IEEE Aerosp. Conf. pp. 1–15 (Mar 2025)

References

[54] Wan, Y., Zhao, W., Zhang, H., Sui, Y., Xu, G., Jin, H.: What do they capture? – a structural analysis of pre-trained language models for source code. In: ICSE. pp. 2377–2388 (May 2022)

[55] Wan, Z., Wei, L.: DeepTriFix: An LLM-based code defect detection framework guided by abstract syntax trees and prompt reasoning. In: AIoTC. pp. 708–713 (Aug 2025)

[56] Wang, K., Pan, B., Feng, Y., Wu, Y., Chen, J., Zhu, M., Chen, W.: XGraphRAG: Interactive visual analysis for graph-based retrieval-augmented generation. In: PacificVis. pp. 1–11 (Apr 2025)

[57] Wang, L., Zhou, Y., Zhuang, H., Li, Q., Cui, D., Zhao, Y., Wang, L.: Unity is strength: Collaborative LLM-based agents for code reviewer recommendation. In: ASE. pp. 2235–2239 (Oct 2024)

[58] Wang, Z., Liu, Y.: XAIpipeline: Automated orchestration of explainable AI services for cloud AI and open-source models. In: SSE. pp. 50–56 (Jul 2025)

[59] Yan, X., Xuan, X., Ono, J.P.P., Guo, J., Mohanty, V., Kumar, S.A., Gou, L., Wang, B., Ren, L.: VISLIX: An XAI framework for validating vision models with slice discovery and analysis. Comput. Graph. Forum **44**(3) (2025)

References

[60] Yang, L., Chen, Y.: Integrating RAG and LLM for automated code review in practice. In: ISCIPT. pp. 591–596 (Sep 2025)

[61] Yue, Z., Yang, J., Li, R., Xiao, Q.: A BIM code compliance checking method based on classification awareness and domain-constrained alignment. In: CBASE. pp. 598–602 (Oct 2025)

[62] Zhang, Y., Chen, S., Fan, L.: A web-based tool for using storyboard of android apps. In: ICSE-Companion. pp. 117–121 (May 2023)

[63] Zhao, J., Sun, Y., Huang, C., Liu, C., Guan, Y., Zeng, Y., Liu, Y.: Towards secure code generation with LLMs: A study on common weakness enumeration. IEEE Trans. Softw. Eng. **51**(12), 3507–3523 (Dec 2025)

[64] Zhou, B., Zhao, H., Wen, Y., Ding, G., Xing, Y., Lin, X., Xiao, L.: Software defect prediction based on semantic views of metrics: Clustering analysis and model performance analysis. Comput. Mater. Contin. **84**(3), 5201–5221 (Jul 2025)

[65] Zhu, S., Zhang, Y., Xu, W., Guo, W., Ye, P.: LLMSDH: An integrated software-assisted development tool based on large language models. In: QRS-C. pp. 671–678 (Jul 2025)

Backup: Search and Selection Details

Database results

Database	Records	Included
IEEE Xplore	43	22
Scopus	68	28
ScienceDirect	15	9
Total	**126**	**59**

Inclusion criteria

- AI/ML/LLM deployment
- explicit explainability component
- SE artifact linkage
- peer reviewed, English, 2019–2025

SMS selection protocol [29, 36, 38].

Backup: Database Search Strings

Search executed **15 January 2026** · 2019–2025 · English · peer reviewed

IEEE Xplore (All Metadata)

```
("explainable AI" OR "XAI" OR "interpretability" OR "model
explanation")
AND ("code generation" OR "code completion" OR "program
synthesis"
  OR "code LLM" OR "large language model")
AND ("software engineering" OR developer OR "code review"

   OR "pull request" OR "test generation")
```

ScienceDirect (sequential refinement)

```
1. "code generation" OR "program synthesis" OR "AI-assisted
programming"
2. AND "explainable AI" OR "interpretable" OR
"interpretability" OR XAI

 3. Filters: 2019--2025 via interface
```

Scopus (String 1: implementation)

```
TITLE-ABS-KEY(("explainable AI" OR "interpretable AI" OR XAI
  OR "model explanation") AND ("code generation" OR
"code completion" OR "program synthesis" OR "large language
model"
OR "code LLM") AND ("software engineering" OR developer OR
"code review" OR "test generation" OR DevOps))

 AND (PUBYEAR > 2018 AND PUBYEAR < 2027)
```

Scopus (String 2: lifecycle)

```
TITLE-ABS-KEY(("explainable AI" OR "interpretable AI" OR XAI
  OR "interpretability") AND ("software development
lifecycle" OR SDLC OR DevOps OR "MLOps" OR "V-Model" OR
"verification and validation") AND (code OR "source code" OR
"test" OR "test case" OR "software system"))

 AND (PUBYEAR > 2018 AND PUBYEAR < 2027)
```

Queries returned **126** initial records (IEEE 43, Scopus 68, ScienceDirect 15); full strings in paper Appendix A [29, 38, 36].

Backup: PRISMA Selection Funnel

PRISMA stage	n	Outcome
Identification	126	IEEE 43, Scopus 68, ScienceDirect 15
After duplicate removal	106	20 duplicates removed
Title/abstract screening	76	30 excluded (outside scope at screening)
Full-text assessment	76	each paper read for empirical XAI–SE integration
Full-text exclusions	17	no XAI technique and/or no SE artifact linkage
Included primary studies	**59**	entered quality coding (Q1–Q3) and taxonomy extraction

Counts match Figure 3 in the paper; selection followed PRISMA 2020 [36, 29].

Backup: Full-Text Assessment Criteria

Assessed in all 76 eligible papers

- **AI deployment:** ML, DL, or LLM used in a software engineering task
- **Explainability:** explicit XAI component, not black-box output only
- **SE artifact linkage:** code, tests, requirements, logs, or pipeline artifacts
- **Empirical evidence:** evaluation, user study, or deployment report

Reasons for exclusion (17 papers)

- no empirical XAI–SE integration at full-text read
- missing explicit explainability technique
- no traceable software engineering artifact
- also screened out: position papers, domain-agnostic works, non-English, pre-2019

Full-text assessment is the gate that turns database hits into codable primary studies.

Inclusion and exclusion rules as stated in Section 3.1 [36, 38].

Backup: Post-Inclusion Quality Coding

Indicator	Yes	Partial	No	Coding rule (59 studies)
Q1 Evaluation reported	59	0	0	explicit metrics, user study, or deployment evaluation
Q2 Faithfulness validated	0	16	43	causal vs. proxy validation (strict causal bar for Yes)
Q3 Persona grounded	8	13	38	concrete stakeholder role operationalised in evaluation

Reliability

20% of studies double-coded; Cohen's $\kappa = 0.81$; disagreements resolved by consensus [31].

Traceability

Six-dimensional taxonomy extraction; per-study matrix in Appendix B; multi-phase studies assigned by dominant task.

Backup: Why This Matters for Applied Intelligent Systems

Governance

Explanations become decision logs: prompts, versions, approvals, retrieval sources, and agent actions.

Quality

Faithfulness checks move from optional post-hoc analysis into acceptance criteria and regression evidence.

Certification

XAI artifacts may help organise governance and software-quality evidence for ISO/IEC 42001 and ISO/IEC 25059 [28, 27].

Boundary condition

The pipeline is a conceptual blueprint from mapped evidence; certification still requires end-to-end industrial validation [18, 25].

Backup: Phase and Model Distribution

Phase	LLM	ML	DL	Q2 Yes	Partial	No	Total
Requirements & Design	5	3	0	0	3	5	8
Implementation & Security	20	3	0	0	6	17	23
Testing & Debugging	8	2	5	0	4	11	15
Maintenance & Operations	4	9	0	0	3	10	13
Total	**37**	**17**	**5**	**0**	**16**	**43**	**59**

Defensible reading

Faithfulness challenges are structurally tied to the LLM-rationale pairing, not only to individual tools [38, 50].

Backup: LLM Explanation Strategies

Strategy	Mechanism	Main limitation
Chain-of-thought	generated rationale with prediction	unverified reasoning path
RAG	retrieve then explain	retrieval and summarisation opacity
Multi-agent	role separation and consensus	latency and non-determinism
Intrinsic probing	internal representation analysis	architecture-specific evidence

Observed in the corpus

CoT and RAG show only marginal partial-faithfulness evidence; multi-agent and intrinsic probing studies show none in the coded sample [40, 63, 57, 54].

Backup: Certification-Ready Artifacts

Requirement type	Pipeline artifact
Governance / accountability	prompts, model versions, approvals, agent actions
Traceability	links from requirements, tests, issues, outputs, explanations
Evaluation evidence	benchmark reports, human studies, acceptance criteria
Runtime assurance	drift alerts, incident reports, rollback records
Provenance	retrieval sources, ranks, snapshots, data and model lineage

Positioning

Certification-ready XAI treats explanations as engineering artifacts, not as decoration around AI outputs [28, 27, 22, 14].

Backup: Study Limitations & Validity

Study limitations

- Three databases only (IEEE, Scopus, ScienceDirect)
- **59/126** after PRISMA — traceable mapping, not census
- Corpus skews to **2024–2025** LLM tools
- Multi-phase studies assigned to one dominant phase

What we still claim

- Reproducible SMS with explicit search strings
- **76** full-text papers assessed before inclusion
- Dual coding with $\kappa = \mathbf{0.81}$
- Directional gaps hold under conservative coding

Limitations bound generalisation; they do not reverse the late-skewed, faithfulness, or persona findings [29, 36, 50].

Backup: Positioning vs Prior Surveys

Lens	Typical prior work	This mapping study
General XAI	taxonomies and metrics	explicit linkage to SE artifacts and lifecycle decisions
AI in SE	broad task coverage	explainability as a first-class coded dimension
LLM tool studies	single-tool or single-phase evidence	cross-phase, cross-model synthesis (ML/DL/LLM)
Arora et al. (COMPSAC 2025)	phase-oriented survey	faithfulness coding (Q2), persona grounding (Q3), certification blueprint

Distinct contribution

An SDLC-wide, persona-aware, faithfulness-coded evidence base for applied intelligent software systems [7, 8, 5].

Backup: Chair Q&A Quick Reference

Likely chair question	Bound in one line	See backup
Is faithfulness coding too strict?	Yes bar = causal validation; **0/59** full, **16** partial; $\kappa = 0.81$	Post-Inclusion Quality Coding
Why only **59** studies?	**126**→**76** full-text →**59**; deliberate scope	PRISMA Funnel
What search strings were used?	Database-specific Boolean queries; **15 Jan 2026**	Database Search Strings
Is the blueprint validated?	No — conceptual synthesis; industrial validation required	Certification-Ready Artifacts
Main study limitations?	Three DBs; corpus skew; mapping not exhaustiveness	Study Limitations
How is this different from Arora et al.?	Adds faithfulness + persona coding and cross-generation synthesis	Positioning vs Prior Surveys
Boxes 1–4 sequential?	No — four **parallel** architecture patterns	Contribution 3 slide